IN THE CLAIMS:

1. (Previously presented) A method for avoiding livelock among two or more input/output (I/O) devices of a computer system comprising a plurality of interconnected processors, one or more shared memories coupled to the processors, and at least one I/O bridge in communicating relationship with the two or more I/O devices, the processors and the one or more shared memories, the method comprising the steps of:

providing at least one coherent buffer and at least one non-coherent buffer at the I/O bridge, the non-coherent buffer coupled to the at least one coherent buffer and to at least one of the I/O devices;

receiving a request for information from a first I/O device that is directly coupled to the I/O bridge;

storing the information requested by the first I/O device in the coherent buffer of the I/O bridge;

receiving a system message at the I/O bridge requesting the information stored in the coherent buffer, the system message originating from other than the first I/O device;

copying at least a portion of the stored information to the non-coherent buffer;

invalidating the stored information within the coherent buffer; and

supplying to the first I/O device at least some of the stored information copied into the non-coherent buffer.

2. (Previously presented) The method of claim 1 wherein

the first I/O device is coupled to the non-coherent buffer by an I/O bus having a bus cycle specifying a predetermined number of bits per I/O bus cycle, and

the quantity of stored information supplied to the first I/O device from the non-coherent buffer corresponds to the predetermined number of bits of one bus cycle.

3. (Original) The method of claim 2 further comprising the steps of:

receiving a second request at the I/O bridge from the first I/O device requesting information;

determining whether the information of the second request is stored in the coherent buffer; and

if the information of the second request is stored in the coherent buffer, supplying at least some of the information to the first I/O device.

4. (Original) The method of claim 3 further comprising the steps of:

if the information of the second request is not stored in the coherent buffer, determining whether the information of the second request is stored in the non-coherent buffer; and

if the information of the second request is stored in the non-coherent buffer, supplying the predetermined number of bits of one bus cycle of the information to the first I/O device.

5. (Previously presented) The method of claim 1 further comprising the steps of:

granting the I/O bridge exclusive ownership relative to the plurality of processors and the other I/O bridges of the computer system over the information stored by the I/O bridge; and

generating an acknowledgement confirming that the stored information has been invalidated by the I/O bridge.

6. (Original) The method of claim 5 further comprising the steps of:

organizing information stored in the one or more shared memories of the computer system into respective cache lines; and

providing one or more cache coherency directories, the one or more cache coherency directories configured to store an ownership status for each cache line,

wherein

the system message requesting information originates from one or more of the directories and the acknowledgement is sent to one or more of the directories.

7. (Previously presented) An input/output (I/O) bridge for use in a distributed shared memory computer system comprising a plurality of interconnected processors and a plurality of shared memories that are coupled to the processors, the I/O bridge configured to provide intercommunication between one or more I/O devices and the plurality of processors or shared memories, the I/O bridge comprising:

at least one coherent buffer configured to store information requested by a first I/O device that is directly coupled to the I/O bridge;

at least one non-coherent buffer coupled to the coherent buffer and to the one or more I/O devices; and

a controller coupled to the coherent buffer and the non-coherent buffer, the controller configured to:

store at least a portion of the information stored in the coherent buffer in the non-coherent buffer in response to receiving a system message originating from other than the first I/O device requesting the information stored in the coherent buffer,

invalidate the information within the coherent buffer, and

supply to the first I/O device, which is directly coupled to the I/O bridge, at least some of the information copied into the non-coherent buffer.

8. (Previously presented) The I/O bridge of claim 7 further wherein

the first I/O device is coupled to the non-coherent buffer by an I/O bus having a bus cycle specifying a predetermined number of bits per I/O bus cycle, and

the quantity of information supplied to the first I/O device from the non-coherent buffer corresponds to ~~is~~ the predetermined number of bits of one bus cycle.

9. (Previously presented) The method of claim 1 further comprising the steps of:

granting the I/O bridge exclusive ownership relative to the plurality of processors and the other I/O bridges of the computer system over the information stored by the I/O bridge; and

following the step of invalidating, generating an acknowledgement confirming that the stored information has been invalidated by the I/O bridge.

10. (Previously presented) The method of claim 1 further comprising the steps of:

organizing information stored in the one or more shared memories of the computer system into respective cache lines; and

providing one or more cache coherency directories, the one or more cache coherency directories configured to store an ownership status for each cache line,

wherein

the system message requesting information originates from one or more of the directories and the acknowledgement is sent to one or more of the directories.

11. (Previously presented) The method of claim 1 further wherein the system message originates from one of the processors or from an I/O device other than the first I/O device.

12. (Previously presented) The method of claim 1 further comprising the steps of:

issuing an initial miss message in response to the system message requesting the information stored in the coherent buffer; and

searching the coherent buffer for the requested information following the issuance of the initial miss message.

13. (Previously presented) The method of claim 12 wherein the steps of copying, invalidating and supplying occur following the searching step.

14. (Previously presented) The method of claim 1 wherein the first I/O device establishes a connection with the I/O bridge in order for the I/O bridge to receive the request from the first I/O device, the method further comprising the step of releasing the connection between the I/O bridge and the first I/O device following the supplying step.

15. (Previously presented) The method of claim 2 wherein the predetermined bits is one of thirty-two and sixty-four bits.

16. (Previously presented) The I/O bridge of claim 7 wherein the at least one coherent buffer and the at least one non-coherent buffer are both disposed at the same buffer having a plurality of entries and a flag for each entry which is set to indicate whether the respective entry is coherent or non-coherent.

17. (Previously presented) The I/O bridge of claim 7 wherein information stored at the at least one coherent buffer is in an exclusive state whether the information is being read or written.

18. (Previously presented) The I/O bridge of claim 7 wherein the at least one coherent buffer participates in a directory based cache coherency protocol executed by the processors.

19. (Previously presented) The I/O bridge of claim 8 wherein the information is organized in terms of cache lines having a plurality of bits and the predetermined number of bits that is supplied to the first I/O device correspond to one of the first thirty-two or first sixty-four bits of the respective cache line.

20. (Previously presented) The I/O bridge of claim 8 wherein the I/O bus is configured to operate in accordance with one of Peripheral Component Interface (PCI), PCI-Extended (PCI-X) and Accelerated Graphics Port (AGP) specification standards.